

《C/C++ 学习 指南》

第19.1讲：面向对象编程的设计思想

作者：邵发 QQ群：417024631

官网：<http://www.afanihao.cn/>

答疑：<http://www.afanihao.cn/kbase/>

C/C++学习指南 邵发 www.afanihao.cn

面向对象编程

面向对象编程是一种**设计思想**，并不局限于语言。

无论是C++，Java，还是C语言，B语言，或者Shell，Perl，Python，甚至汇编语言，都可以实现面向对象编程。

“面向对象编程”是设计思想，C, C++是实现手段。

（相对来说，C++更容易实现这种设计思想，而C则相对是麻烦了一点。）

面向对象编程

如何实现面向对象编程？步骤如下：

第一步：假设存在一个对象。初步设想它应该提供哪些服务。
(把这个对象当成一个工具、或者是一个服务员)

第二步：定义如何使用它的服务。细化为函数。

- (1) create/destroy: 创建与销毁对象
- (2) 其他功能函数: 指该对象提供的服务

第三步：选择一种实现方法，写代码完成上述函数接口。

实例：DataStore

现在的需求是：需要存储一系列Student对象。

第一步：定义服务

假设存在一个对象DataStore，它为我们提供Student对象的存储服务器。它提供以下的服务：

- (1) 可以向它加入一个对象
- (2) 可以按ID来查找一个对象
- (3) 可以按ID删除一个对象
- (4) 可以打印显示所有的对象

实例：DataStore

第二步：细化为函数

定义一个类型

```
struct DataStore  
{  
};
```

(注：成员变量先不用考虑)

创建与销毁

```
DataStore* ds_create();  
void ds_destroy(DataStore* store);
```

实例：DataStore

第二步：细化为函数

其他功能函数：

- (1) 可以向它加入一个对象
`void ds_add(DataStore* store, const Student* obj);`
- (2) 可以按ID来查找一个对象
`Student* ds_find(DataStore* store, int id);`
- (3) 可以按ID删除一个对象
`void ds_remove(DataStore* store, int id);`
- (4) 可以打印显示所有的对象
`void show_all(DataStore* store);`

C/C++学习指南 邵发 www.afanihao.cn

实例：DataStore

第二步：细化为函数

确定该对象的使用方式

```
// 创建一个对象
DataStore* store = ds_create();

// 调用ds_add
Student obj;
ds_add(store, &obj);

// 销毁对象
ds_destroy(store);
```

C/C++学习指南 邵发 www.afanihao.cn

实例：链表对象

至此，已经完成了面向对象编程的设计工作。

够用

刚好够用

分析

1. DataStore是一种类型，用户可以创建多个DataStore，来存储不同的数据。每个对象之间互不影响。
2. 使用者只关心如何使用，并不关心内部的实现。